Return-Path: <shen@Think.COM>
Date: Thu, 16 Apr 92 12:39:58 EDT
From: Tracy Shen <shen@Think.COM>
Sender: shen@Think.COM
To: jonathan@Think.COM
Cc: barbara@quake.think.com, brewster@Think.COM
In-Reply-To: Jonny Goldman's message of Tue, 14 Apr 92 18:12:41 PDT
<9204150112.AA05918@philo.quake.think.com.>
Subject: releasing seeker

> Date: Tue, 14 Apr 92 18:12:41 PDT
> From: Jonny Goldman <jonathan@Think.COM>
>
> Barbara,
>
> I tried to build the software, and it's by no means ready for prime time.
>
> Tracy,
>
> The seeker directory has lots of test stuff. Please identify what's for
> the release and what's not. If you can, create a directory called
> seeker-release with the files necessary for release, and a Makefile the can
> build them (you'll see a directory call /proj/wais/test/seeker, feel free
> to us it as your seeker-release).
> If you could go over the instructions on how to run it too...

It will save you trouble and time next time you are going to build a
release, let me know when and what you need as I usually use
the seeker directory as my working directory simply because I need disk
space.
One thing important to point out is that seeker works with b5 but not b4.
If you try to build it with the wais b4 , it either won't run or
will break when executing the indexer or the server.

I can't make much use of the test/seeker directory as I don't have the
right permission to move files in the directory around.
I will use latest/seeker instead and make sue the instructions are
included. Anything else we need to have the release built?

> Brewster,
>
> I strongly recommend testing the software a little while before we announce
> it.

I always have doubt about the usefulness of the seeker software.
Users with small amount of data, the serial search engine is enough to them.
It probably looks more appealing to people who have large amount of
data to index. Unfortunately, the seeker does not seem robust when
building big database as the CMDRS does - mainly because of
the loose rule we have in selecting words and pair words,
and keeping the hash table in memory which could grow pretty quickly.

It works against seeker algorithm to keep words in the database generously.
CMDRS only keep words that account 60% of the total database word
occurrences.
I now even doubt about this strategy after I analyzed the queries from
real DowQuest users. The prilimiary result shows that
most words in the queries are high frequency words.
A pretty high percentage of them are even non-terms. They together don't
form a proper noun, but express most ideas users want to know. CMDRS
does not handle this kind of query well unless the headline contains

the non-term or the high-frequency-term words. In another words, it heavily depends on having appropriate headline description of the stories themselves. My point is usuing the word frequency as the major criteria to choose searchable terms does not turn out to be as effective in searching as we expected when we designed the CMDRS for DowQuest.

- Jonny G